Latent Dirichlet Allocation and Singular Value Decomposition based Multi-Document Summarization

Rachit Arora
Computer Science and Engineering
Indian Institute of Technology Madras
Chennai - 600 036, India.
rachitar@cse.iitm.ernet.in

Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras
Chennai - 600 036, India.
ravi@cse.iitm.ernet.in

Abstract

Multi-Document Summarization deals with computing a summary for a set of related articles such that they give the user a general view about the events. One of the objectives is that the sentences should cover the different events in the documents with the information covered in as few sentences as possible. Latent Dirichlet Allocation can break down these documents into different topics or events. However to reduce the common information content the sentences of the summary need to be orthogonal to each other since orthogonal vectors have the lowest possible similarity and correlation between them. Singular Value Decomposition is used to get the orthogonal representations of vectors and representing sentences as vectors, we can get the sentences that are orthogonal to each other in the LDA mixture model weighted term domain. Thus using LDA we find the different topics in the documents and using SVD we find the sentences that best represent these topics. Finally we present the evaluation of the algorithms on the DUC 2002 Corpus multi-document summarization tasks using the ROUGE evaluator to evaluate the summaries. Compared to DUC 2002 winners, our algorithms gave significantly better ROUGE-1 recall measures.

1 Introduction

The task of Multi-Document Summarization consists of computing the summary of a set of related documents of a corpus such that they cover the major details of the events in the documents. Let us say we have a set of M related documents together in a corpus. These documents all share a central theme or event which is the topic on which the documents are based on. We say that this event is the property of all the documents i.e. is the property of the corpus. The documents have other sub-events or topics which might be common between some of the documents, which support or give more details regarding the central theme. In essence these topics revolve around this central event and are linked to it by explaining the cause,effect,statistics etc of the central event. Thus together the central theme and the sub-events form the *topics* for the set of documents. In the following article we have used the term *topic* and *events* interchangeably.

One way of approaching the task of multi-document summarization is to break the documents into these topics and then describe or represent these topics adequately in the summary. Thus we can view the documents as being composed of topics, which we have to infer, and the visible variables are the words of the documents. Words are interpreted as means of expressing topics a document is composed of.

Latent Dirichlet Allocation (LDA) [3] is a generative three-level hierarchical Bayesian probabilistic model for collections of discrete data such as text documents. The documents are modeled as a finite mixture over an underlying set of topics which, in turn, are modeled as an infinite mixture over an underlying set of topic probabilities. Thus in the context of text modeling, the topic probabilities provide an explicit representation of the documents.

One way of interpreting the LDA model is that it breaks down the collection of documents into independent *topics* by representing the document as a mixture of topics using a probability distribution. The topics in turn are represented as a mixture of words using a probability distribution. Another way of looking at it is that LDA soft-clusters the words of the documents into these topics i.e. instead of hard clustering and assigning a word to one topic, it gives a probability of the word belonging to the topic. Thus in a way we can view the these documents as a three level hierarchical Bayesian model with the topics, their distribution and the Dirichlet parameters as latent variables and words and documents that they belong to as the only visible variables.

Using LDA we infer the topics that the documents are

composed of. A sentence of the document can be viewed as representing a single topic,multiple topics or maybe connecting different topics of a document or a set of documents. Here we are dealing with extraction based multi-document summarization i.e. we are extracting the entire sentence from the document without any modification to it like removal or adding of words and combining sentences together in a summary.

Singular Value Decomposition can be used to find the orthogonal representations of vectors. The entry with the highest value in a singular vector of the SVD transformed matrix corresponds to the original vector which has the greatest impact in this orthogonal direction. Thus the original vector is the best representative of this new orthogonal direction. Using this we can find vectors which have the greatest impact in different orthogonal directions and call them as orthogonal representations as the vectors are not orthogonal to each other (however the transformed vectors are). The orthogonal representations have the lowest possible similarity and correlation among them, thus are useful in lowering redundancy in the system. In this case the vectors are sentences of the documents weighted by the topic mixture model of LDA. Thus using LDA we find the different topics in the documents and using SVD we find the sentences that best represent these topics.

In the past multi-document summarization algorithms have been mostly about word alignment on the summaries, by using the term frequency and inverse-document frequency or some combination of other weighting mechanisms of the words in the documents. The sentences are then given measures using the weights of the words in some combination or other and using some similarity and anti-redundancy techniques [5] [13]. These weighting mechanisms give limited choice in terms of giving weights to the words and the sentences. Other multi-document summarizers have taken a probabilistic approach by using mixture models [11].

On the other hand LDA, even though it was meant as a text-topic model, has found wide application in the field of image and video processing to capture objects and actions [6] [14]. LDA has found limited application in the fields of Information Retrieval and Statistical Natural Language Processing. SVD has been extensively used in the field of Principal Component Analysis [12] for the purpose of query retrieval.

In this we propose a novel approach of using LDA to capture the topics that the documents are based on and using SVD to find the most orthogonal representations of these topics as sentences in the documents. Using LDA we represent these documents as being composed of topics and use that as a central idea in forming the term by sentence matrix for SVD. Then using SVD we extract the most sentences that best represent these topics by obtaining the most orthogonal representations and forming the multi-document summary. Whereas other summarization algorithms weight the sentences *without capturing* the events, we weight the sentences by capturing these events that the documents are based on by using LDA. Also LDA, SVD and the summarization algorithm based on it assume the documents to be "bag-of-words" and we don't involve the grammar. Thus the approach is purely statistical and the summarization algorithm doesn't involve the structure of the documents or of the sentences in terms of grammar and the meanings conveyed by the words.

In Section 2 we present the **Latent Dirichlet Allocation and Singular Value Decomposition based Multi-Document Summarization** algorithm and explain the intuition and reasons behind using LDA and SVD separately and then combining them. Section 3 gives the multi-document summarization algorithm in detail and its working. Section 4 gives the evaluation of the algorithm on the DUC 2002 Corpus task of multi-document summarization and Section 5 talks about the future work in this direction. We have used the limitations and the work-arounds in the LDA model we faced in the multi-document summarization algorithm presented in [1]

2 LDA-SVD based Multi-Document Summarization

Using LDA we can break down the documents into topics using a mixture model. Thus each Document D_k is a mixture model over the Topics T_j given by the probability distribution P($T_j|D_k$) and each Topic T_j is a mixture model over the Word W_i of the vocabulary given by the probability distribution P($W_i|T_j$). Considering the sentences of the documents, if we have K number of topics, we can have K different independent representation for each sentence, assuming that a sentence represents only one topic. They are independent because under the Dirichlet distribution, the topics are independent of each other.

Thus we can represent the entire corpus in K independent representations in the form of a matrix $A^{(j)}$ for the topic T_j formed in the following way :

- For each sentence S_{kr} belonging to the document D_k form its sentence vector for the topic T_j in the following manner :
  - If the word $W_i \in S_{kr}$, its value in the sentence vector is given by
    $$S^{(j)}_{kri} = P(W_i|T_j) * P(T_j|D_k) \ \forall W_i \in S_{kr} \quad (1)$$
  - For a word $W_i \notin S_{kr}$, its value in the sentence vector is 0.
    $$S^{(j)}_{kri} = 0 \ \forall W_i \notin S_{kr} \quad (2)$$

- Thus we get a sentence vector $S_{kr}^{(j)} = [S_{kri}^{(j)}]^T$ where i ranges over the entire vocabulary for the sentence S_{kr} at the Topic T_j.

- Thus we get the matrix $A^{(j)} = [S_{kr}^{(j)}]$, for $1 \leq k \leq M$ and $1 \leq r \leq R_k$, where R_k is the number of sentences in the document k

Dropping the super-script (j) which stands for the topic T_j for convenience (i.e. we implicitly assume that the topic is T_j in the following discussion), the Singular Value Decomposition of a m $\times$ n matrix A, where without loss of generality m $>$ n, can be defined as :

$$A = U\sigma V^T \tag{3}$$

- U = $[u_{ij}]$ is an m $\times$ n column-orthonormal matrix whose columns are called left singular vectors.

- σ = diagonal(σ_1, σ_2, ... , σ_n) is an n $\times$ n diagonal matrix whose diagonal elements are non-negative singular values sorted in descending order

- V = $[v_{ij}]$ is an n $\times$ n orthonormal matrix whose columns are called right singular vectors.

- If rank(A) = r, then σ satisfies $\sigma_1 > \sigma_2 > ..... > \sigma_r > \sigma_{r+1} = ... = \sigma_n = 0$

The interpretation of applying the SVD to the LDA mixture model weighted terms by sentences matrix A can be made from two different view-points :

- From transformation point of view, the SVD derives a mapping between the multi-dimensional space spanned by the term vectors representing the topic T_j and the r-dimensional singular vector space with all of its axes linearly-independent. This mapping projects each column vector i of the matrix A, which is the representation of the sentence i being projected as representing a single topic T_j, to the column vector $v_i = [v_{i1}\ v_{i2}\ ...\ v_{ir}]^T$ of matrix V^T and maps each row vector k of the matrix A, which tells the occurrence of the term k as belonging to the Topic T_j in each of the sentences, to the row vector $u_k = [u_{k1}\ u_{k2}\ .\ .\ .\ u_{kr}]$ of matrix U. Here each element v_{ix} of v_i, u_{jy} of u_j is called the index with the x^{th} and the y^{th} singular vectors, respectively.

- From semantic point of view, the SVD derives the latent semantic structure from the sentence represented by matrix A [12]. This operation reflects a breakdown of the original sentence matrix representing the Topic T_j into r linearly-independent base vectors or concepts. Each term and sentence from the original sentences is jointly indexed by these base vectors/concepts under the Topic T_j.

Symbol	Meaning
D_k	The k^{th} Document
S_r	The r^{th} Sentence
T_j	The j^{th} Topic
W_i	The i^{th} Word
M	The number of documents
R	The number of sentences
K	The number of Topics
$P(W_i\|T_j)$	Probability of Word W_i given the Topic T_j
$P(T_j\|D_k)$	Probability of Topic T_j given the Document D_k

Table 1. List of Symbols

A unique SVD feature which is lacking in conventional IR technologies is that the SVD is capable of capturing and modeling interrelationships among terms so that it can semantically cluster terms and sentences.

Furthermore, as demonstrated in [2], if a word combination pattern is salient and recurring in a document, this pattern will be captured and represented by one of the singular vectors. The magnitude of the corresponding singular value indicates the importance degree of this pattern in the topic T_j. Any sentences containing this word combination pattern will be projected along this singular vector, and the sentence that best represents this pattern will have the largest index value with this vector.

3 The Algorithm

- Decompose the set of documents into individual sentences based on the fact that sentences end with a full-stop "." . We take care of cases where the full stop might be for short forms as in Sgt., George W. Bush etc. We also identify the sentences which are parts of a speech and combine these sentences into one. Thus we form the candidate sentence set SEN.

- We apply Porter Stemmer to stem the words to their root form and remove the stop words using a standard stop-words list.

- Using LDA find the probability distributions of the documents over the topics i.e. $P(T_j|D_k)$ and of the topics over the vocabulary i.e. $P(W_i|T_j)$.

- For each topic T_j construct a term by sentence matrix $A^{(j)}$ in the following way :

  - For each sentence S_{kr} belonging to the document D_k form its sentence vector for the topic T_j in the following manner :

input : A Corpus of Documents
output: S - The set of sentences in the summary

$S \leftarrow 0$;
Apply LDA on the Corpus of Documents;
Let A and V be 3-D Matrices;;
for $j \leftarrow 1$ **to** K **do**
  $k \leftarrow 0$;
  for $d \leftarrow 1$ **to** M **do**
    R_d *- Number of Sentences in the Document* D_d;
    for $r \leftarrow 1$ **to** R_d **do**
      for $i \leftarrow 1$ **to** V **do**
        if $W_i \in S_{dr}$ **then**
          A[j][i][k] = P($W_i|T_j$)*P($T_j|D_d$)*P(D_d);
        end
        else
          A[j][i][k] = 0;
        end
      end
      k++;
    end
  end
  V[j]← SVD(A[j]);
end
Let PT be a vector that stores the topics in the decreasing order of probability calculated by Equation 6;
Let R be a vector of size K initialized to 1;
for $n \leftarrow 1$ **to** N **do**
  m ← PT[n];
  r ← R[m];
  $v_{xr} \leftarrow$ *Maximum value in the Singular Vector* v_r *of the matrix V[m]*;
  $S_x \leftarrow$ *Sentence corresponding to* v_{xr};
  if $S_x \notin S$ **then**
    S← S ∪ S_x;
  end
  R[m]++;
  if *n = K* **then**
    n← 1;
  end
end

Algorithm 1: LDA-SVD Algorithm

  - $*$ If the word $W_i \in S_{kr}$, its value in the sentence vector is P($W_i|T_j$)*P($T_j|D_k$)*P(D_k)

$$S^{(j)}_{kri} = P(W_i|T_j) * P(T_j|D_k) \ \forall W_i \in S_{kr} \quad (4)$$

  - $*$ For a word $W_i \notin S_{kr}$, its value in the sentence vector is 0.

$$S^{(j)}_{kri} = 0 \ \forall W_i \notin S_{kr} \quad (5)$$

  - Thus we get a sentence vector $S^{(j)}_{kr} = [S^{(j)}_{kri}]^T$ where i ranges over the entire vocabulary for the sentence S_{kr} at the Topic T_j.
  - Thus we get the matrix $A^{(j)} = [S^{(j)}_{kr}]$, for $1 \leq k \leq M$, $1 \leq r \leq R_k$, where R_k is the number of sentences in the document k.
  - Since the SVD of a matrix is independent of the order of its column vectors i.e. the sentence vectors in case of matrix $A^{(j)}$, we can arrange the sentences of all the documents in any order.

- Apply the Singular Value Decomposition on the term by sentence matrix weighted by LDA mixture model i.e. the matrix $A^{(j)}$ to get the new sentence matrix $V^{(j)}$.
- Repeat this procedure for all the topics i.e. for $1 \leq j \leq K$, form the matrix $A^{(j)}$ and apply SVD on it to get the matrix $V^{(j)}$.
- In the right singular vector matrix $V^{(j)}$, each sentence i is represented by the column vector $v_i = [v_{i1}\ v_{i2} \ldots v_{ir}]^T$.
- Calculate the probability of topic T_j according to

$$P(T_j) = \sum_{k=1}^{M} P(T_j|D_k) * P(D_k) \quad (6)$$

  and form a vector PT which stores the topics in the decreasing order of P(T_j).
- Set n to 0 and let us say that the number of sentences required in the summary is N.
- Select the topic m corresponding to the n^{th} value of the vector PT.
- In the matrix $V^{(m)}$, select the sentence which has the largest index value in the r^{th}_m singular vector i.e. let the largest value in the r^{th}_m column of the matrix $V^{(m)}$ be v_{xr_m} and select the x^{th} sentence of the original matrix $A^{(m)}$. If the sentence is already in the summary, ignore it, otherwise include it in the summary set S.

- Increment r_m by 1 and n by 1. If n equals to the number of sentences required in the summary i.e. N, finish the operation. If n reaches the number of Topics i.e. K, reset n to 0 and continue the operation.

- At the end we will be left with a set of summary sentences S, which we need to order to represent a readable summary. Arrange the sentences of S in the relative order that they appear in their respective documents. If their is a tie, break it arbitrarily.

4 Evaluation

For the purpose of evaluation of our multi-document summarization algorithms we used the DUC 2002 Corpus dataset. The data was made up of 59 sets of Documents each containing on an average 10 documents and the candidate algorithms were required to make multi-document summary for each document set. The length of the summary was limited to 200 and 400 words. The candidate algorithms were supposed to be extraction based i.e. the sentences in the summary were supposed to be as they were in the documents, without any sort of modification. In addition for each document set we are given 2 model summaries against which the extracted candidate summary could be compared against.

We used the ROUGE Evaluator [8] which evaluates the summary using Ngram Co-Occurrence Statistics [9] and using Longest Common Subsequence and Skip-Bigram Statistics [10]. We have calculated the ROUGE scores separately for 200 and 400 length summary as we want to even see the effect of the length of the summary on the quality of the summary. We are mainly interested in the ROUGE-1 Recall score, which uses unigram statistics, since the precision scores can be manipulated by adjusting the length of the candidate summary [9]. Also since there were 2 model summaries for each document set, we have used the average score for each document set.

In the ROUGE settings we have used Porter Stemmer to stem the words to their root form in the computed summary and the model summaries given. We compare the results of the multi-document summarization algorithm against the top two algorithms of the DUC2002 Multi-Document Summarization task, "Generating Single and Multi-Document Summaries with GISTEXTER" (GISTEXTER) [5] and "Writing Style Recognition and Sentence Extraction" (WSRSE) [13] in terms of ROUGE-1 recall measures. We also take a look at the 95% confidence interval. We have also looked at another similar work that uses only LDA to construct the summary [1].

We have evaluated the LDA-SVD multi-document summarization algorithm by considering both cases of removing stop-words and not removing stop-words from the computed and the model summaries. Table 2 tabulates the ROUGE-1 recall values and its 95% confidence interval.

From the table we see that the LDA-SVD algorithm for computing multi-document summaries performs much better than both the DUC 2002 winners (GISTEXTER and WSRSE). Also the lower bound of the 95% confidence interval for the ROUGE-1 recall measure for **LDA-SVD** algorithm is higher than the upper bound of the 95% confidence interval of both GISTEXTER and WSRSE, thus showing that the LDA-SVD multi-document summarization algorithm is statistically better. This holds for the summaries of length 200 and 400 words, thus showing that the algorithms works irrespective of the size of the summary to be computed.

Compared to using only LDA to construct the summary as given in [1], we notice that the combined LDA-SVD model presented in this article gives better result. This can be attributed to the feature that the algorithm presented in this article uses the SVD model to reduce the redundancy in the system. This is evident in the fact that although the improvement in summary of length 200 words is just 0.9% for the LDA-SVD model over the LDA model, the summary for length of 400 words shows an improvement of 1.9%. This can be explained because with more number of words for the target summary, there is a greater chance for redundancy in the system. The SVD part of the LDA-SVD algorithm is able to prevent including this redundancy in the summary which the algorithm in [1] using only LDA is not able to do.

5 Conclusion and Future Work

In this we have shown a novel way of approaching the task of multi-document summarization by using LDA combined with SVD. We can use the mixture-models to explicitly represent the documents as topics or events and use these topics as the basis of forming sentences of the documents. Then using SVD we can get the most orthogonal representation of these sentences which forms the basis of choosing the sentences from the documents to form the summary. The performance of this approach on the DUC 2002 Multi-Document Summarization tasks shows statistically significant improvement over other summarization algorithms in terms of the ROUGE-1 recall measures.

We can extend the basic idea presented in this article by replacing LDA with other topic models like "Pachinko allocation: DAG-structured mixture models of topic correlations" (PAM) [7] and "Mixtures of hierarchical topics with Pachinko allocation" (HPAM) [4], in which we can estimate the number of topics as given in [1], or use it with "Hierarchical Dirichlet Processes" (HDP) [16] and "Nonparametric Bayes Pachinko Allocation" (NBP) [15] in which the number of topics is inferred within the model.

ROUGE Setting	GISTEXTER	WSRSE	LDA	LDA-SVD
Length = 200	0.48671	0.48694	0.55613	0.56107
StopWords Kept	0.46198 - 0.51153	0.46000 - 0.51294	0.54065 - 0.57279	0.54591 - 0.57701
Length = 200	0.39526	0.40060	0.45625	0.45874
StopWords Removed	0.36437 - 0.42729	0.37202 - 0.42910	0.43533 - 0.47703	0.43919 - 0.47902
Length = 400	0.56327	0.58006	0.60775	0.61937
StopWords Kept	0.54737 - 0.57991	0.55579 - 0.60231	0.59683 - 0.61868	0.60870 - 0.63087
Length = 400	0.46669	0.48511	0.50198	0.51238
StopWords Removed	0.44647 - 0.48815	0.45952 - 0.51206	0.48571 - 0.51752	0.49765 - 0.52792

Table 2. Results. Above Value is the Recall Measure for ROUGE-1 and Below Value is its 95% Confidence Interval

Even though the task or the application we have considered here is Multi-Document Summarization, the favorable results for LDA-SVD algorithm show that it is possible to combine the two techniques of SVD (from PCA) and LDA (from SNLP), not only for computing multi-document summaries but even in the field of Information Retrieval like finding similarity between documents and query retrieval. In fact instead of using the normal Term Frequency and Inverse Document Frequency to weight the matrix in SVD and other PCA methods, we can use LDA to weight them by using its mixture model.

LDA gives another frame of reference or a dimension to SVD and other PCA techniques by introducing the concept of topics. By capturing these topics we can represent the terms in a more distinctive and more importantly meaningful fashion which greatly enhances the scope of finding meaningful singular vectors in SVD, as shown by the LDA-SVD based multi-document summarization algorithm and its success over the other term-frequency based algorithms.

References

[1] R. Arora and R. Balaraman. Latent dirichlet allocation based multi-document summarization. *Proceedings of the Second Workshop on Analysis for Noisy Unstructured Data AND*, 2008.

[2] D. S. T. Berry Michael W and O. G. W. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of machine Learning Research 3*, pages 993–1022, 2003.

[4] W. L. David Mimno and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. *Proceedings of the 24th international conference on Machine learning*, 24:633–640, 2007.

[5] S. M. Harabagiu and F. Lacatusu. Generating single and multi-document summaries with gistexter. *In Proceedings of the DUC 2002*, pages 30–38, 2002.

[6] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:524–531, 2005.

[7] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. *Proceedings of the 23rd international conference on Machine learning*, 23:577–584, 2006.

[8] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*, 2004.

[9] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. *Proceedings of the Human Technology Conference 2003 (HLT-NAACL-2003)*, 2003.

[10] C.-Y. Lin and F. J. Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004.

[11] S. R. M. Saravanan and B. Ravindran. A probabilistic approach to multi-document summarization for generating a tiled summary. *Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications*, 6:167 – 172, 2005.

[12] T. K. L. G. W. F. Scott C. Deerwester, Susan T. Dumais and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[13] H. van Halteren. Writing style recognition and sentence extraction. *In U. Hahn and D. Harman (Eds.), Proceedings of the workshop on automatic summarization*, pages 66–70, 2002.

[14] X. Wang and E. Grimson. Spatial latent dirichlet allocation. *Proceedings of Neural Information Processing Systems Conference (NIPS) 2007*, 2007.

[15] A. M. Wei Li and D. Blei. Nonparametric bayes pachinko allocation. *Proceedings of Conference on Uncertainty in Artificial Intelligence*, 2007.

[16] M. B. Y.W. Teh, M.I. Jordan and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association,*, 101(476):1566–1581, 2006.